

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-132543

(43)Date of publication of application : 10.05.2002

(51)Int.Cl.

G06F 11/34

G06F 9/46

(21)Application number : 2000-332108

(71)Applicant : HITACHI LTD

(22)Date of filing : 25.10.2000

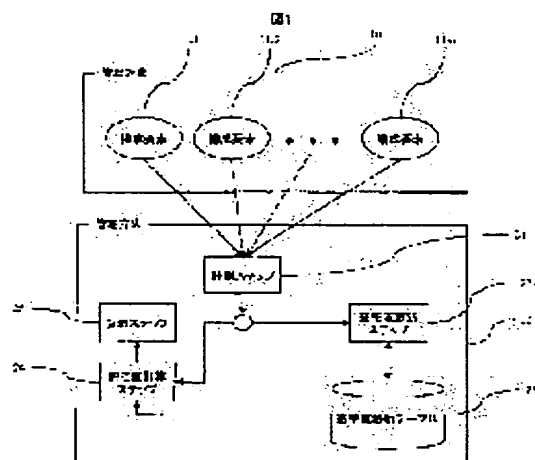
(72)Inventor : TAKANO HIDEKI  
 AKATSU MASA HARU  
 KURODA SAWAKI  
 SONOMURA TOSHIHIRO  
 FUJII NAOHIRO

## (54) CONTROL METHOD OF COMPUTER SYSTEM

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To resolve the problems that in a computer system, when controlling a control object, although if threshold is set to an operation of the control object, abnormality of the object can be detected, it is difficult to specify which configuration element of the object has a problem to be resolved, and if thresholds are set to operations of each configuration element, in some cases, abnormality of configuration elements are detected even though there is no problem in the whole of the operation of the object.

**SOLUTION:** In a control method of computer system, a reference value, which is the value to be measured in a normal operation of the elements of the object, is provided. When operating the object, each configuration element of the object is measured to acquire deviations by comparing the reference value with measured value. By displaying in orderly sequence according to the deviations, points of the objects with high possibility of irregularity in the object are noticed to a manager.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2002-132543  
(P2002-132543A)

(43) 公開日 平成14年5月10日 (2002. 5. 10)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テマコード (参考)
G 0 6 F 11/34		G 0 6 F 11/34	S 5 B 0 4 2
9/46	3 3 0	9/46	3 3 0 C 5 B 0 9 8
	3 4 0		3 4 0 B

審査請求 未請求 請求項の数 7 O L (全 18 頁)

(21) 出願番号 特願2000-332108(P2000-332108)

(22) 出願日 平成12年10月25日 (2000. 10. 25)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 ▲高▼野 英樹

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 赤津 雅晴

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(74) 代理人 100075096

弁理士 作田 康夫

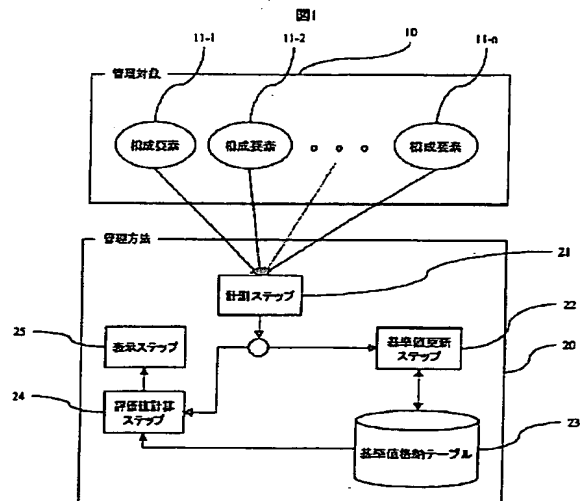
最終頁に続く

(54) 【発明の名称】 計算機システムの管理方法

(57) 【要約】

【課題】 計算機システムにおいて、管理対象を管理する時、その管理対象の挙動に対してしきい値を設ければ、管理対象の異常は検出できるが、管理対象の構成要素のどれが問題なのかを特定することは難しい。逆に、各構成要素の挙動に対してしきい値を設けた場合、管理対象全体の挙動は問題ないのに、構成要素の異常は検出される場合がある。

【解決手段】 管理対象の構成要素が、正常動作時に計測されるべき値である基準値を設ける。管理対象稼動時に、管理対象の各構成要素を計測し、基準値と計測値とを比較してずれを求める。ずれに応じて順序付けて表示することで、管理対象内の異常である可能性の高い個所を管理者に知らせる。



## 【特許請求の範囲】

【請求項1】 計算機システムにおける、複数の構成要素から成る管理対象に対して、その要素の挙動を計測し、前記挙動の基準値を設定し、計測値と前記基準値を比較し、前記計測値と前記基準値のずれに応じて、当該複数の要素と前記計測値、及び前記基準値を順序付けて表示することを特徴とする計算機システムの管理方法。

【請求項2】 計算機システムにおける、管理対象全体の挙動を計測し、前記管理対象の挙動が異常であると判断するためのしきい値を設定し、前記管理対象の計測値としきい値を比較し、前記管理対象の前記計測値が前記しきい値を超えていた場合に、前記管理対象の構成要素の前記計測値と前記基準値のずれに応じて、当該複数の要素と前記計測値と前記基準値を順序付けて表示することを特徴とする計算機システムの管理方法。

【請求項3】 前記構成要素はプログラム内手続きであり、前記挙動は処理時間であることを特徴とする請求項1及び2記載の計算機システムの管理方法。

【請求項4】 前記計測値の履歴の平均を前記基準値として設定することを特徴とする請求項1及び2記載の計算機システムの管理方法。

【請求項5】 前記ずれとして偏差(計測値-基準値)を利用することを特徴とする請求項1及び2記載の計算機システムの管理方法。

【請求項6】 前記ずれとして比率(計測値/基準値)を利用することを特徴とする請求項1及び2記載の計算機システムの管理方法。

【請求項7】 前記管理対象の計測の際に、前記管理対象の構成要素の挙動の計測値を受け取った時に計測値格納テーブルに保存し、管理対象全体の挙動の計測値がしきい値を超えなかった時に計測値格納テーブルからその管理対象の構成要素に関する計測値を削除することを特徴とする請求項1及び2記載の計算機システムの管理方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 計算機システムの管理方法であって、管理対象の異常発生時に管理対象を構成するどの要素が問題であるかを通知する異常検出に関する。特に、複数の計算機システム上の複数のプログラムが強調して動作する分散アプリケーションにおいて、性能異常発生時に異常個所を示す異常検出に関する。

## 【0002】

【従来技術】 計算機システムに関する管理方法であって、管理対象を構成する複数の要素から問題を検出するために、各要素にしきい値を設定し、計測値がしきい値を超えた場合に通知することで、管理対象を管理することが特開2000-105707に示されている。また、コンピュータ・アソシエイツ株式会社のUnicenter(R) TNG(TM) Response Manager(TM) Optionでは、しきい値を自動的

に求めることが可能である。一方、Microsoft(R)の基本ソフトウェアWindowsNT(R) Workstation 4.0では、タスクマネージャによりリソースの使用状況を確認でき、負荷のかかっているプロセスを判別することで問題のありそうなプログラムを把握できる。

## 【0003】

【発明が解決しようとする課題】 計算機システムの要素である管理対象を管理する時、その管理対象の挙動に対してしきい値を設ければ、管理対象の異常は検出できるが、管理対象の構成要素のどれが問題なのかを特定することは難しい。逆に、管理対象の各構成要素の挙動に対してしきい値を設けた場合、管理対象全体の挙動は問題ないのに、構成要素の異常は検出される場合がある。異常個所特定のためにWindowsNT(R)のタスクマネージャを使用すれば、リソースの使用状況が分かるが、それが正常であるのか問題であるのかが分からない。本発明は、管理対象内の異常個所を探す手がかりを提供することを目的とする。

## 【0004】

【課題を解決するための手段】 本発明の計算機システムの管理方法では、管理対象を構成する複数の要素に対して、基準値を設ける。基準値は、基準値格納テーブルに格納する。計測ステップでは、管理対象を構成する要素を計測する。基準値更新ステップでは、計測値を用いて、基準値格納テーブル内の基準値を必要に応じて更新する。評価値計算ステップでは、基準値と計測値を比較して、ずれを求める。表示ステップでは、ずれに応じて順序付けて表示することで、管理対象内の異常である可能性の高い個所を管理者に知らせる。

【0005】 (課題を解決するための手段2) また、本発明の他の管理方法では、管理対象を構成する要素に対して基準値を設ける。基準値は基準値格納テーブルに格納する。管理対象に対してもしきい値のような管理対象の異常を判定する基準となる値を設ける。異常判定ステップでは、管理対象を計測し、管理対象の基準値と計測値を比較して管理対象が異常であるかを判定する。計測ステップでは管理対象の構成要素を計測して、計測値格納テーブルに格納する。基準値更新ステップでは、計測値を用いて、基準値格納テーブル内の基準値を必要に応じて更新する。異常判定ステップで、管理対象が異常だと判定された場合、評価値計算ステップにより、異常だと判定した管理対象の計測値と関連のある構成要素の計測値を計測値格納テーブルから取得し、基準値を比較してずれを求める。表示ステップではずれに応じて順序付けて表示する。管理対象が異常であった時、その構成要素の中から異常である可能性の高い個所を管理者に知らせる。

## 【0006】

【発明の実施の形態】 図1に本発明の実施例1のシステムの構成図を示す。計算機システムにおいて、管理対象

10を構成する複数の要素11に対して、基準値を設ける。基準値は、基準値格納テーブル23に格納する。計測ステップ21では、管理対象10を構成する要素11を計測する。基準値更新ステップ22では、計測値を用いて、基準値格納テーブル23内の基準値を必要に応じて更新する。評価値計算ステップ24では、基準値と計測値を比較して、ずれを求める。表示ステップ25では、ずれに応じて順序付けて表示することで、管理対象内の異常である可能性の高い個所を管理者に知らせる。

【0007】図2に他のシステムの構成図を示す。管理対象10を構成する要素に対して基準値を設ける。基準値は基準値格納テーブル23に格納する。管理対象10に対してもしきい値のような管理対象の異常を判定する基準となる値を設ける。異常判定ステップ31では、管理対象10を計測し、管理対象10の基準値と計測値を比較して管理対象10が異常であるかを判定する。計測ステップ32では管理対象10の構成要素11を計測して、計測値格納テーブル33に格納する。基準値更新ステップ22では、計測値を用いて、基準値格納テーブル23内の基準値を必要に応じて更新する。異常判定ステップ31で、管理対象10が異常だと判定された場合、評価値計算ステップ24により、異常だと判定した管理対象の計測値と関連のある構成要素11の計測値を計測値格納テーブル33から取得し、基準値を比較してずれを求める。表示ステップ25ではずれに応じて順序付けて表示する。管理対象10が異常であった時、その構成要素11の中から異常である可能性の高い個所を管理者に知らせる。

【0008】図3で示した複数のプロセスA、B、Cが協調して動作するアプリケーション50に本発明を適用して、アプリケーション50の応答が通常より遅かった時に最も影響の大きかった処理を検出する場合を例として説明する。プロセスAは、ユーザ40からの要求を受けて手続き $\alpha$ 1を開始する。プロセスAはプロセスBの手続き $\beta$ 1を要求し、次に手続きB2を要求する。手続きB2ではプロセスCの手続き $\gamma$ 1を要求する。

【0009】このアプリケーション50の異常を検出するための構成を図4に示す。アプリケーション50の各プロセスには、処理時間を計測するために管理エージェント60を付加する。管理エージェント60で計測した処理時間などの計測値は、処理時間通知70として管理サーバ80に送る。管理サーバ80では処理時間の基準値と計測値のずれを計算しずれに応じて順序付けて手続きとその計測値、基準値などを表示する。

【0010】管理エージェント60の各ステップについて説明する。処理時間計測ステップ61は、各手続きの開始時刻、他手続き呼び出し時刻、他手続きからの復帰時刻、終了時刻を計測し、計測用テーブル62を利用して処理時間を計算する。処理時間とは、他の手続きを呼び出している時間を除いて、その手続きのみに要した時間である。計測値送出ステップ63は、管理サーバ80に処理時

間通知70を送出する。

【0011】管理サーバ80の各ステップについて説明する。計測値受信ステップ81は、管理エージェント60から処理時間通知70を受信する。基準値更新ステップ82は、処理時間と基準値格納テーブル83に格納された基準値を使用して基準値を更新し、基準値格納テーブル83に格納する。本例では、手続き毎に要した時間の平均を基準値として使用する。評価値計算ステップ84は、処理時間と基準値とを比較して基準値からのずれを求める。本例では、偏差を求める。表示ステップ85は、ずれを表示して管理者に知らせる。

【0012】アプリケーション50のプロセスBの手続き $\beta$ 2の処理時間を管理エージェント60が計測し、管理サーバ80に送出する処理の流れを説明する。手続き $\beta$ 2では、まず手続き $\beta$ 2のスレッドIDと処理名を得る。処理名は、プロセス名と手続き名を含む。スレッドIDは、手続き $\beta$ 2が同時に複数の要求を処理する場合、各計測値を区別できるようにするために必要である。

【0013】次に、手続き $\beta$ 2の開始時刻を取得するためにスレッドIDと処理名を引数としてstartを呼ぶ。これで、手続き $\beta$ 2の処理をするための準備が終わったので、次に手続き $\beta$ 2の本来の処理を行う。手続き $\gamma$ 1を呼ぶ時は、手続き $\gamma$ 1の呼び出し時刻を取得するためにスレッドIDと処理名を引数としてsuspendを呼ぶ。手続き $\gamma$ 1から復帰したら復帰時刻を取得するためにスレッドIDと処理名を引数としてresumeを呼ぶ。手続き $\beta$ 2の本来の処理が終了したらスレッドIDと処理名を引数としてendを呼ぶ。

【0014】start、suspend、resume、endの処理内容を説明する。startでは、引数としてスレッドIDと処理名を得る。計測用テーブル62に引数として得たスレッドIDと処理名を格納する。次に、計測用テーブル62のスレッドIDと処理名が一致する個所の一時停止時刻と停止時間に0を格納する。現在時刻を得て、計測用テーブルのスレッドIDと処理名が一致する個所の開始時刻に現在時刻を格納する。suspendでは、引数としてスレッドIDと処理名を得る。現在時刻を得て、計測用テーブル62のスレッドIDと処理名が一致する個所の一時停止時刻に現在時刻を格納する。resumeでは、引数としてスレッドIDと処理名を得る。計測用テーブル62のスレッドIDと処理名が一致する個所の一時停止時刻と停止時間を得る。現在時刻から一時停止時刻を引いた値を停止時間に足す。算出した値を計測用テーブル62のスレッドIDと処理名が一致する個所の停止時間に格納する。endでは、引数としてスレッドIDと処理名を得る。計測用テーブル62のスレッドIDと処理名が一致する個所から開始時刻と停止時間を得る。現在時刻から開始時刻と停止時間を引いて処理時間を求める。管理サーバ80に現在時刻と処理名と処理時間を含んだ処理時間通知70を送出する。

【0015】次に管理サーバ80がアプリケーション50の

各手続きに要した時間を受けとってから管理者に表示するまでの流れを説明する。管理サーバ80は、計測値受信ステップ81で、管理エージェント60から時刻と処理名と処理時間が送出されるのを待つ。受信すると基準値更新ステップ82により、受信した処理時間を利用して基準値の更新を行う。

【0016】次に、評価値計算ステップ84により基準値と処理時間のずれを計算する。最後に、ずれに応じて手続きとその処理時間、基準値などを表示する。基準値更新ステップ82について説明する。引数として処理名と処理時間を得る。基準値格納テーブル83に引数で得た処理名が存在しないなら、基準値格納テーブル83に引数として渡された処理名、更新時間に現在時刻、基準値に処理時間、更新回数に1を格納する。基準値格納テーブル83に引数で得た処理名が存在するなら、基準値格納テーブル83から処理名をキーにして基準値と更新回数を得る。 $(\text{基準値} \times \text{更新回数} + \text{処理時間}) \div (\text{更新回数} + 1)$ を計算して新しい基準値を計算する。基準値格納テーブル83の処理名が一致する個所の更新時間に現在時刻、基準値に新しい基準値、更新回数に更新回数+1を格納する。

【0017】評価値計算ステップ84について説明する。引数として、処理名と処理時間を得る。基準値格納テーブル83から処理名をキーにして基準値を得る。処理時間から基準値を引いて偏差を求める。戻り値として、偏差を返す。図5に表示ステップ85が表示する画面の例を示す。ずれを算出するたびに新たに行を作成し、ずれに応じて適当な個所に追加表示する。画面では、偏差の降順に表示している。時刻、処理名、処理時間、基準処理時間、偏差に対して昇順、降順に表示することも可能である。図6にグラフ表示の例を示す。偏差の基準値からのずれを棒グラフ表示している。四角で濃く示された個所は、これまでにアプリケーション50を計測した履歴の中で最も基準値よりずれた時の偏差を表している。

【0018】上記の実施例によれば、アプリケーション50の挙動が異常であった場合、実施例1の表示結果を管理者に提供することにより、管理者に異常な手続きを特定する手がかりを与えることが可能である。

【0019】実施例1において、管理エージェント90と管理サーバ110の通信量を減らすことを目的としたシステムを例にとって実施例2を説明する。図7は、そのシステムの構成図である。管理エージェント90内でアプリケーション50の各手続きの処理時間の基準値からのずれを計算し、管理サーバ110ではずれに応じて順序付けで表示する。start処理、suspend処理、resume処理は実施例1と同じである。end処理は、処理時間を求めるところまでは実施例1と同じであるが、その後、基準値更新ステップ82により基準値を更新し、評価値計算ステップ91によりずれを計算する。引数は、処理名と処理時間である。ずれを求めたら、評価値通知100として管理サーバ110に送出する。管理サーバ110の処理の流れを説明

する。評価値受信ステップ111でずれを受け取ったら、表示ステップ112で表示する。

【0020】上記の実施例によれば、実施例1の効果に加えて、管理サーバ110と管理エージェント90間の通信量が実施例1に比べて少ないため、管理サーバ110と管理エージェント90が別計算機で稼動することがあっても、ネットワークの通信量が少なくて済む。

【0021】実施例1において、ずれとして比率を利用し、さらに、一覧表示では既に処理名が表示されていた場合にその処理名の行を削除してから新たに行を追加する方法を利用する実施例3を示す。ずれとして比率を利用した時の評価値計算ステップ91の処理の流れを説明する。基準値格納テーブル83から、引数で与えられた処理名をキーにして基準値を得て、引数で与えられた処理時間を基準値で割って比率を得てその比率を戻り値として返す。

【0022】図8に一覧表示の例を示す。この表示方法では、ずれを受け取った時に、既に同じ処理名のずれが表示されていたら、その表示の行を削除してから、新しく受け取ったずれをずれに応じて適当な位置に表示する。

【0023】図9にグラフ表示の例を示す。手続き毎に比率を棒グラフ表示している。四角で濃く示された個所は、これまでにアプリケーション50を計測した履歴の中で最も基準値よりずれた時の比率を表している。

【0024】上記の実施例によれば、ずれとして、比率を利用することで、基準値からのずれをずれの割合で知ることができる。一覧表示の表示方法の変更では、ある処理名は必ず表中に一つしか存在しないため、容易に現在の処理状況を知ることができる。

【0025】CORBAを基盤として構築されたアプリケーションに適用して、アプリケーションを管理対象とし、アプリケーションの応答時間にしきい値を設けて、しきい値より応答時間が遅かった時に、そのセッション内の影響の大きかった手続きを特定する場合を例にとって実施例4を説明する。セッションとは要求が発生するたびに呼ばれる一連の手続きである。

【0026】図10で示すように管理対象は、オブジェクト $\alpha$ 、 $\beta$ 、 $\gamma$ の3つのプログラムで構成されているアプリケーション120である。オブジェクト $\alpha$ 、 $\beta$ 、 $\gamma$ はそれぞれ計算機A、B、Cで稼動している。

【0027】図11にアプリケーション120を管理するシステムの構成を示す。オブジェクト $\alpha$ 、 $\beta$ 、 $\gamma$ には管理エージェント130を配する。管理エージェント130は、アプリケーション120の各手続きの処理時間、経過時間を計測する。計測結果は、処理時間通知140として、管理サーバ150に送出する。管理サーバ150では、処理時間通知140を受信したら、計測値格納テーブル153に格納する。基準値格納テーブル83内の基準値を処理時間の計測した処理時間を利用して更新する。全体経過時間が管理

者が前もって設定したしきい値を超えていないかを確認し、超えている場合、超えていたセッションの処理時間と基準値のずれをずれに応じて表示する。

【0028】管理エージェント130の各ステップについて説明する。処理ID付与ステップ131は、アプリケーション120に対する要求が発生するたびにそのセッションの処理に対して処理IDを付与する。処理IDは、“計算機名.オブジェクト名.処理名.呼び出された時刻”などから構成される。処理名は、オブジェクト名と処理名を含んでいる。処理時間計測ステップ132は、各手続きの開始時刻、他手続きの呼び出し時刻、他手続きからの復帰時刻、終了時刻を得て、計測用テーブル133を利用して処理時間と経過時間を算出する。計測値送出ステップ134は、時刻、呼び出し元処理ID、処理ID、処理時間、経過時間から構成される処理時間通知140を管理サーバ140に送出する。

【0029】次に管理サーバ150の各ステップについて説明する。全体しきい値入力ステップ151は管理者が、手続き0を要求してから手続き0が終了するまでの経過時間に要しても良い時間のしきい値を設定するステップである。計測値受信ステップ152は、管理エージェントから各処理時間通知140を受信して、計測値格納テーブル153に格納する。基準値更新ステップ82は、処理時間を使用して基準値格納テーブル83に格納された基準値を更新するステップである。基準値として計測値の履歴の平均値を使用する。

【0030】全体経過時間比較ステップ154は、管理者が入力したしきい値と実際に全体でかかった経過時間を比較する。比較の結果、しきい値を超えていなかった場合、処理IDをたどって計測値格納テーブル153からそのセッションで出た計測値を全て削除する。評価値計算ステップ155は、全体経過時間比較ステップ154を実行した結果、全体の経過時間が管理者の設定したしきい値を超えていた場合、計測値格納テーブル153内の処理IDをたどって処理時間と各手続きの基準値とを比較してずれを計算するステップである。ずれは、比率で表す。時刻、処理名、処理時間、基準値、比率は、表示情報格納テーブルに格納する。表示ステップ157は、表示情報格納テーブル156に保存された情報を表示するステップである。

【0031】アプリケーション120の各オブジェクトの手続きは、管理エージェント130の機能と呼ぶことにより、開始時刻、他手続きの呼び出し時刻、他手続きからの復帰時刻、終了時刻を通知する。これらの通知処理は管理サーバ150に各手続きの処理時間と経過時間を報告するために必要である。オブジェクトβの手続き1の処理の流れで説明する。

【0032】手続き1は、手続き0から呼び出されるが、この時、手続き0の処理IDを得る。処理ID付与ステップと呼ぶことにより、手続き1の処理IDを得る。処理名と

して“オブジェクトβ”、“手続き1”を得る。処理IDを引数としてstartと呼ぶ。これで、手続き1の処理を開始する準備は完了である。手続き1の本来の処理を行う。手続き2と呼ぶ前には、処理IDを引数としてsuspendと呼ぶ。手続き2から復帰したら、処理IDを引数としてresumeと呼ぶ。手続き1の本来の処理をして最後に処理IDを引数としてendと呼ぶ。

【0033】start、suspend、resume、endは、実施例1のstart、suspend、resume、endで、スレッドIDと処理名を処理IDに置き換えたものと同じである。endの処理では、さらに現在時刻から開始時刻を引いて経過時間を算出、管理サーバには、現在時刻、呼び出し元処理ID、処理ID、処理時間、経過時間を処理時間通知140として送出する。

【0034】管理サーバ150が処理時間通知140を受け取って基準値を更新するまでの処理の流れを説明する。計測値受信ステップ152では、処理時間通知140を受け取るのを待つ。受け取ると基準値更新ステップ82により、基準値を更新する。

【0035】図12に管理サーバ150が異常な個所を表示する処理の流れをPADで示す。計測値受信ステップ152により、管理エージェント130から処理時間通知130を受信するのを待つ。受信したら、受信した経過時間に対応するしきい値が設定されているかを調べ、設定されているなら、全体経過時間比較ステップ154で経過時間と管理者が設定したしきい値を比較する。経過時間がしきい値を超えていたら、評価値計算ステップ155により、処理時間と基準値の間の比率を計算し、表示ステップ157によりずれの大きさに応じて表示する。各ステップについて説明する。基準値更新ステップ82は実施例1と同じである。

【0036】図13(a)に全体経過時間比較ステップ154のPADを示す。引数として、処理ID、しきい値、全体経過時間を得る。全体経過時間がしきい値を超えていないなら、計測値格納テーブルからの削除154-1を行い、しきい値を超えていないことを通知する。全体経過時間がしきい値を超えているならしきい値を超えたことを通知する。

【0037】図13(b)に計測値格納テーブルからの削除154-1の処理の流れをPADで示す。計測値格納テーブル153から処理IDが一致する項目を削除し、削除した処理IDを記録しておく。計測値格納テーブル153内に記録した処理IDが呼び出し元処理IDである項目が存在する間、以下の処理を行う。計測値格納テーブル153から呼び出し元処理IDが記録した処理IDである項目の処理IDを得る。計測値格納テーブルからの削除154-1と呼ぶ。

【0038】図14(a)に評価値計算ステップ155のPADを示す。引数として最初の手続きの処理IDを得る。比率計算155-1と呼ぶ。図14(b)に比率計算155-1の処理の流れを示す。引数として処理IDを得る。計測値格納テー

ブル153から処理IDが引数で渡された処理IDである処理時間と時刻を得る。処理IDを記録しておく。基準値格納テーブル83から処理名をキーにして基準値を得る。ここでの処理名は、処理IDに含まれるものである。処理時間と基準値から比率を計算する。表示情報格納テーブル156に時刻、処理IDに含まれる計算機名、処理IDに含まれる処理名、処理時間、基準値、比率を格納する。計測値格納テーブル153の呼び出し元処理IDが記録した処理IDである項目が存在する間、以下の処理を繰り返す。計測値格納テーブル153の呼び出し元処理IDが記録した処理IDである項目の処理IDを得る。比率計算155-1を呼ぶ。図15に表示ステップ157の表示例を示す。この表示では、比率が大きい順にソートして表示している。

【0039】上記の実施例によれば、アプリケーション120の挙動が異常であった場合のみ、ずれの表示を行うため、管理者に余分な情報を提供せず、効率良く管理することを可能とする。

【0040】実施例4において、管理エージェント160に全体経過時間としきい値の比較、基準値更新、ずれの計算機能を配し、管理サーバ200では、各管理エージェント160からのずれの収集、表示機能を配したアプリケーション120を管理することで管理エージェント160、管理サーバ200間の通信量を減らすシステムを例にとって実施例5を説明する。

【0041】図16にシステムの構成図を示す。管理エージェント160では、処理時間、経過時間を計測し、計測値格納テーブル164に格納する。さらに、基準値格納テーブル83内の基準値の更新を行う。計測した全体経過時間と管理者が設定したしきい値の比較を行い、管理サーバ200に比較結果通知170を出す。管理サーバ200は比較結果通知170を受け取ると、しきい値を超えていたなら、各管理エージェント160に評価値要求通知180を出す。各管理エージェント160は、評価値要求通知180を受け取ると、計測値格納テーブル164に格納された処理時間と基準値格納テーブル83内に格納された基準値からずれを計算し、評価値通知190として、管理サーバ200に送出する。管理サーバ200では、評価値通知190を受け取り、ずれに応じて表示する。

【0042】各ステップについて説明する。セッションID付与ステップ161では、セッション毎にIDを付与する。処理時間計測ステップ162では、計測用テーブル163を利用して各手続きの処理時間と経過時間を計測する。計測した値は、計測値格納テーブル164に格納する。基準値更新ステップ82では、計測値を利用して基準値格納テーブル83内の基準値を更新する。

【0043】全体経過時間比較ステップ165では、管理者が全体しきい値入力ステップ151により入力したしきい値と、全体経過時間(ここでは、手続き0の経過時間)を比較する。比較結果は、比較結果通知170として管理サーバ200に送出する。管理サーバ200の比較結果受信ス

テップ201で比較結果通知170を受信する。評価値要求送出ステップ202では、全体経過時間がしきい値を超えていた場合、全ての管理エージェント160に評価値要求通知180を送出する。管理エージェント160の評価値要求受信ステップ167では、評価値要求通知180を受け取る。

【0044】評価値計算ステップ168では、計測値格納テーブル164に格納されたセッションIDが一致する処理時間と基準値格納テーブル83の基準値とを比較してずれを計算する。ずれとして比率を利用する。評価値送出ステップ169により、評価値通知190を管理サーバ200に送出する。管理サーバ200の評価値受信ステップでは、評価値通知190を受け取ると、表示情報格納テーブル156に格納する。表示ステップ157では、表示情報格納テーブル156内の情報をずれに応じて表示する。

【0045】斜線で示した箇所は、全体経過時間を計測する手続きにのみ付加するステップである。ここでは、斜線で示したステップは手続き0に付加する。手続き0に付加した後の手続き0の処理の流れを説明する。まず、セッションID付与ステップ161により、セッションを一意に識別できるIDを得る。

【0046】次に、セッションIDを引数としてstartを呼ぶ。手続き0の本来の処理をする。手続き1を呼ぶ前に、セッションIDを引数としてsuspendを呼ぶ。手続き1を呼ぶ時は、セッションIDを受け渡す。手続き1から復帰したらセッションIDを引数としてresumeを呼ぶ。手続き0の本来の処理をした後、セッションIDを引数としてendを呼んで終了する。手続き0から呼ばれる手続き1の処理の流れを説明する。手続き0から引数としてセッションIDを受け取り、手続き0と同様に処理する。手続き2を呼ぶ時にも手続き0から受け取ったセッションIDを受け渡す。start、suspend、resume処理は、実施例4の処理IDをセッションIDに置き換えたものと同じである。

【0047】手続き0に付加するend処理を説明する。以下、処理IDとは、計算機名、処理名を含んだものである。実施例4のend処理と同様に、処理時間、経過時間を計算したら、セッションID、処理ID、処理時間、経過時間を計測値格納テーブル164に格納する。次に、基準値更新ステップ82により、基準値を更新する。基準値として計測値の履歴の平均を利用する。全体経過時間比較ステップ165により、全体経過時間(手続き0の経過時間)と管理者が設定したしきい値を比較して比較結果を求める。最後に、比較結果送出ステップにより、比較結果を比較結果通知170として管理サーバに送出する。手続き0以外の手続きに付加するend処理では、全体経過時間比較ステップ165と比較結果送出ステップ166は行わない。

【0048】全体経過時間比較ステップ165の処理の流れを説明する。引数としてしきい値と全体経過時間を得る。しきい値と全体経過時間を比較して、しきい値の方

が大きければしきい値を超えていないという通知を返す。全体経過時間の方が大きければ、しきい値を超えたという通知を返す。

【0049】管理サーバ200では、比較結果受信ステップ201により、比較結果通知170を得て評価値要求送出ステップ202を行う。図17に評価値要求送出ステップ202の処理の流れを示す。比較結果が、全体経過時間がしきい値を超えていることを示しているなら、全ての管理エージェント160に対して評価値要求通知180を送出する。

【0050】管理エージェントは、評価値要求受信ステップ167により、評価値要求通知180を受け取ると、評価値計算ステップ168により、ずれを計算し、評価値送出ステップ169により、ずれを評価値通知190として管理サーバ200に送出する。評価値計算ステップ168の処理の流れを説明する。引数としてセッションIDと処理IDを得る。計測値格納テーブル164から、セッションIDと処理IDが一致する個所の処理時間を得る。基準値格納テーブル83から処理名をキーにして基準値を得る。処理時間を基準値で割って比率を求める。時刻、セッションID、処理ID、処理時間、基準値、比率を返す。これらの値を評価値通知190として管理サーバ200に送出すると、評価値受信ステップ203で受け取って、表示情報格納テーブル156に格納される。表示ステップ157では、格納された値を表示する。

【0051】上記の実施例によれば、実施例4の効果に加えて、管理サーバ200と管理エージェント160間の通信量が実施例4に比べて少ないため、管理サーバ200と管理エージェント160が別計算機で稼動することがあっても、ネットワークの通信量が少なくて済む。

【0052】実施例5において、管理エージェントに各計算機、各オブジェクト、各手続きの状態を計測する手段を配し、処理時間の計測を開始する時に手続きと、その手続きが含まれる計算機、その手続きが含まれるオブジェクトの状態を計測し、計測した結果を一覧表示と関連付けて表示することで管理者が問題追求するための多くの手がかりを提供することを可能とする実施例6のシステムの例である。

【0053】図18は、実施例5の図16に変更追加する構成要素を示している。処理時間計測ステップ162と表示ステップ157と比較結果受信ステップ201は、処理時間計測ステップ211と表示ステップ231と比較結果受信ステップ233に変更し、評価値要求送出ステップ202は表示ステップ231から呼び出されるように変更し、それ以外は追加要素である。

【0054】各計算機の状態として、CPU使用率、メモリ使用量、プロセス数を計測する。各オブジェクトの状態として、処理しているリクエスト数を計測する。各手続きの状態として、処理しているリクエスト数を計測する。

【0055】計算機の状態であるCPU使用率、メモリ使

用量、プロセス数は、OSから取得可能であるとする。処理時間計測ステップ211は、処理時間計測ステップ162の機能に加えて、end処理内で、CPU使用率、メモリ使用量、プロセス数をOSから得て、計算機状態格納テーブル215に格納する機能を持つ。

【0056】オブジェクトの状態である処理しているリクエスト数を計測する方法を説明する。管理エージェント210毎に、オブジェクト状態格納テーブル216を配する。さらに、処理しているリクエスト数を計測するためにオブジェクト毎にカウンタを、オブジェクトに一つずつ配する。オブジェクト状態計測ステップ213では、次のようにリクエスト数を計測する。処理時間の計測を開始するために呼ぶstartで、処理しているリクエスト数のカウンタに1を加える。処理時間の計測を終了する時に呼ぶendでは、処理しているリクエスト数のカウンタから1を引く。さらに、引数で受け取ったセッションIDと処理IDと時刻をオブジェクト状態格納テーブル216に格納する。オブジェクト状態格納テーブル216内の処理しているリクエスト数にはカウンタの値を書き込む。これらの情報が必要な時は、セッションID、時刻、処理IDをキーにして、検索する。

【0057】手続きの状態である、処理しているリクエスト数を計測する方法を説明する。管理エージェント210に、手続き状態格納テーブル217を配する。さらに、処理しているリクエスト数を計測するためにカウンタを手続きに一つずつ配する。手続き状態計測ステップ214では、この手続き状態格納テーブル217と各カウンタに対して、start、endで、オブジェクト状態格納テーブルに対して行う処理と同様の処理を行うことで、処理しているリクエスト数を計測する。

【0058】アプリケーション120の全体経過時間が管理者が設定したしきい値を超えた場合、比較結果通知170が比較結果受信ステップ233に届く。比較結果受信ステップでは、届いた比較結果通知170を比較結果格納テーブル234に格納する。表示ステップ231では、比較結果格納テーブル234に格納された情報を表示する。

【0059】図19に、表示される画面の遷移の様子を示す。図中の矢印は、画面の遷移を表している。比較結果格納テーブル234に格納された情報をしきい値を超えた要求一覧240として表示する。ここで、セッションIDが87521を選択すると、表示ステップ231から評価値要求ステップ202を呼び出し、管理エージェント210から評価値を得て、処理時間一覧表241を表示する。処理時間一覧表241で、管理者が計算機名、オブジェクト名、手続き名を選択することでそれぞれの状態を表示する。各項目が選択されると、管理サーバ230の周辺状況収集ステップ232で、その項目のセッションIDと時刻と処理IDを含む、計算機状態要求、オブジェクト状態要求、手続き状態要求の何れかを管理エージェントに出す。管理エージェント210の計算機状態計測値受信ステップ218、オブ



ジェクト状態計測値要求受信ステップ、手続き状態計測値要求受信ステップでは、それを受け取るとセッションIDと時刻と処理IDをキーにして、計算機状態格納テーブル215、オブジェクト状態格納テーブル216、手続き状態格納テーブル217の何れかから計測値を取り出す。管理エージェント210には、計算機状態計測値送信ステップ、オブジェクト状態計測値送信ステップ、手続き状態計測値受信ステップを配し、計算機状態格納テーブル215、オブジェクト状態格納テーブル216、手続き状態格納テーブル217から計測値の平均を算出し、基準値とする。セッションIDと時刻と処理IDをキーにして取り出した計測値を基準値で割って比率を計算する。計算機状態通知、オブジェクト状態通知、手続き状態通知の何れかとしてセッションID、時刻、処理ID、計測値、基準値、比率を管理サーバに送出する。管理サーバ230の表示ステップ231では、それらの情報を、計算機Cの状態242、計算機CのオブジェクトAの状態243、計算機CのオブジェクトA、手続き2の状態244として表示する。

【0060】さらに、各表示内の気になる項目を選択することで、その項目に関連する周辺の状態を表示する。例えば、計算機Cの状態242の画面中のCPU使用率を選択すると、管理サーバ230内の周辺状況収集ステップ232では、CPU使用率収集要求を出し、管理エージェント210内の計算機状態計測値要求受信ステップ218でその要求を受け取り、計算機状態計測値送出ステップ219では、CPU使用率と時刻の一覧を得て、管理サーバ230へ送出する。表示ステップ231では、それらをCPU使用率のグラフ表示245として表示する。図中の白丸245-1が、選択する個所に対応する部分である。

【0061】以上の機能を用いて、管理者が問題追求を行う一例を説明する。アプリケーション120の応答時間が、管理者が設定したしきい値を超えていた場合、しきい値を超えた要求一覧表210に項目が追加される。管理者は、セッションIDが87521の情報を見たい場合、そのセッションの行を選択すると、どの手続きで時間がかかったのかを示す、処理時間一覧表示241が出る。この場合、計算機Cで稼動しているオブジェクトAの手続き2が最も通常より時間がかかったことが分かる。

【0062】そこで、どうして時間がかかったのかを調べるために、管理者は、計算機Cを選択する。選択すると、計算機Cの状態242が現れる。比率を見ることで、通常より異常であるのがCPU使用率であることが分かる。念のために、計算機CのオブジェクトAの状態243、計算機CのオブジェクトA、手続き2の状態244を見てみると、通常とあまり変わらないことが分かる。そこで、計算機のCPU使用率が原因であることが分かる。さらに、CPU使用率のグラフ表示245を得ると、CPU使用率が急に上がっている個所245-2があることが分かる。その個所245-2で問題が発生している可能性が高いと予想できるので、次に、その個所245-2を選択すると、該当するセッション

の処理時間一覧表が表示される。管理者、次に、そのセッションについて問題があるかを追跡していくことになる。

【0063】上記の実施例によれば、アプリケーションの処理速度が遅かった時、遅かった手続きを遅かった順に表示するのに加え、各手続きが呼ばれた時の計算機、オブジェクト、手続きの状態を管理者に関連付けて表示することにより、異常個所の追求をより迅速に行うことを可能にする。

【0064】

【発明の効果】計算機システムの構成要素である管理対象が異常である時、本発明を用いれば、基準値と計測値のずれによって管理対象を構成する要素名を表示することができ、管理者は表示を手がかりに異常個所を探すことができる。

【図面の簡単な説明】

【図1】管理対象内の異常個所を検出するシステムの構成図である。

【図2】管理対象が異常であった場合に管理対象内の異常個所を検出するシステムの構成図である。

【図3】複数のプロセスで構成された管理対象の例である。

【図4】管理エージェントから計測値を送出して処理時間を管理するシステムの構成図である。

【図5】ずれを計算するたびに表に追加する一覧表示例である。

【図6】偏差のずれのグラフ表示例である。

【図7】管理エージェントからずれを送出して処理時間を管理するシステムの構成図である。

【図8】ずれを計算する毎に既に値があればその行を削除してから新たに行を追加する一覧表示例である。

【図9】比率のずれのグラフ表示例である。

【図10】CORBAで実現された管理対象の例である。

【図11】管理エージェントから計測値を送出してセッション毎に処理時間を管理するシステムの構成図である。

【図12】管理エージェントから計測値を送出してセッション毎に処理時間を管理する時の異常個所表示処理の流れである。

【図13】管理エージェントから計測値を送出してセッション毎に処理時間を管理する時の全体経過時間比較ステップの処理の流れである。

【図14】管理エージェントから計測値を送出してセッション毎に処理時間を管理する時の評価値計算ステップの処理の流れである。

【図15】セッション毎に処理時間を管理する時の管理対象の一覧表示例である。

【図16】管理エージェントからずれを送出してセッション毎に処理時間を管理するシステムの構成図である。

【図17】管理エージェントからずれを送出してセッシ

ョン毎に処理時間を管理する時の評価値要求送出ステップの処理の流れである。

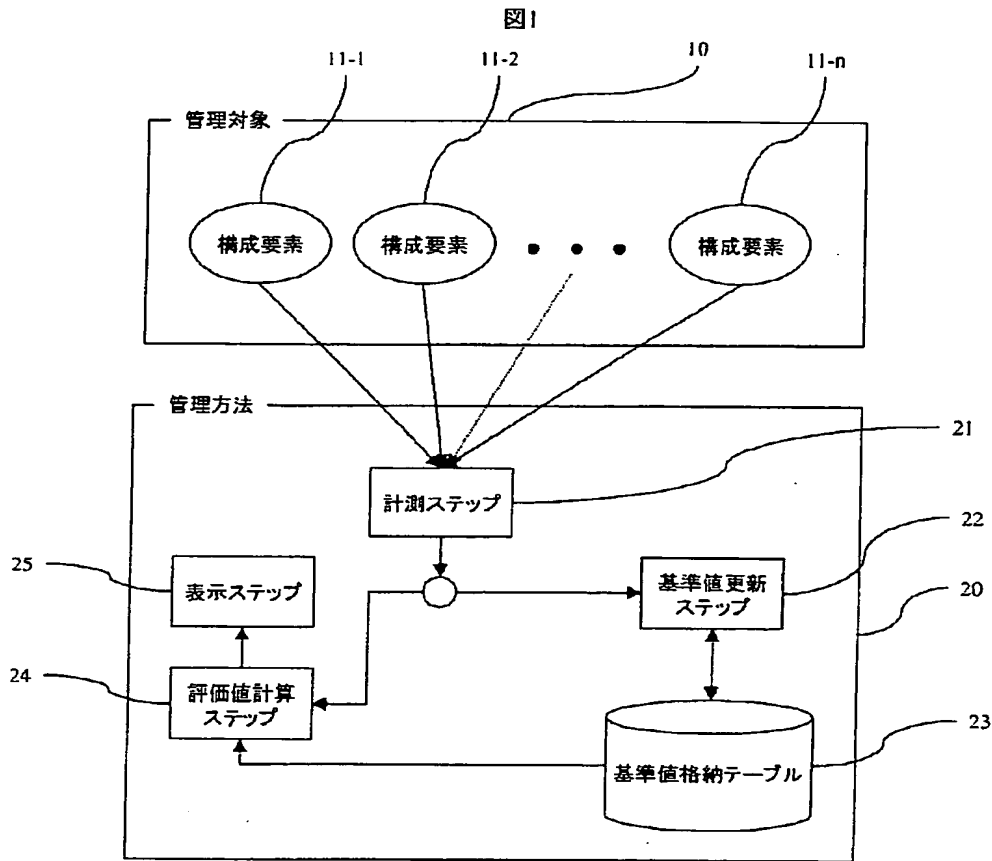
【図18】処理時間のずれと周辺状況を関連付けて表示した時の例である。

【図19】管理対象内の異常個所を検出するシステムの構成図である。

【符号の説明】

10…管理対象、11…構成要素、20…管理方法、21…計測ステップ、22…基準値更新ステップ、23…基準値格納テーブル、24…評価値計算ステップ、25…表示ステップ、30…管理方法、31…異常判定ステップ、32…計測ステップ、33…計測値格納テーブル、40…ユーザ、50…アプリケーション、60…管理エージェント

【図1】



【図5】

図5

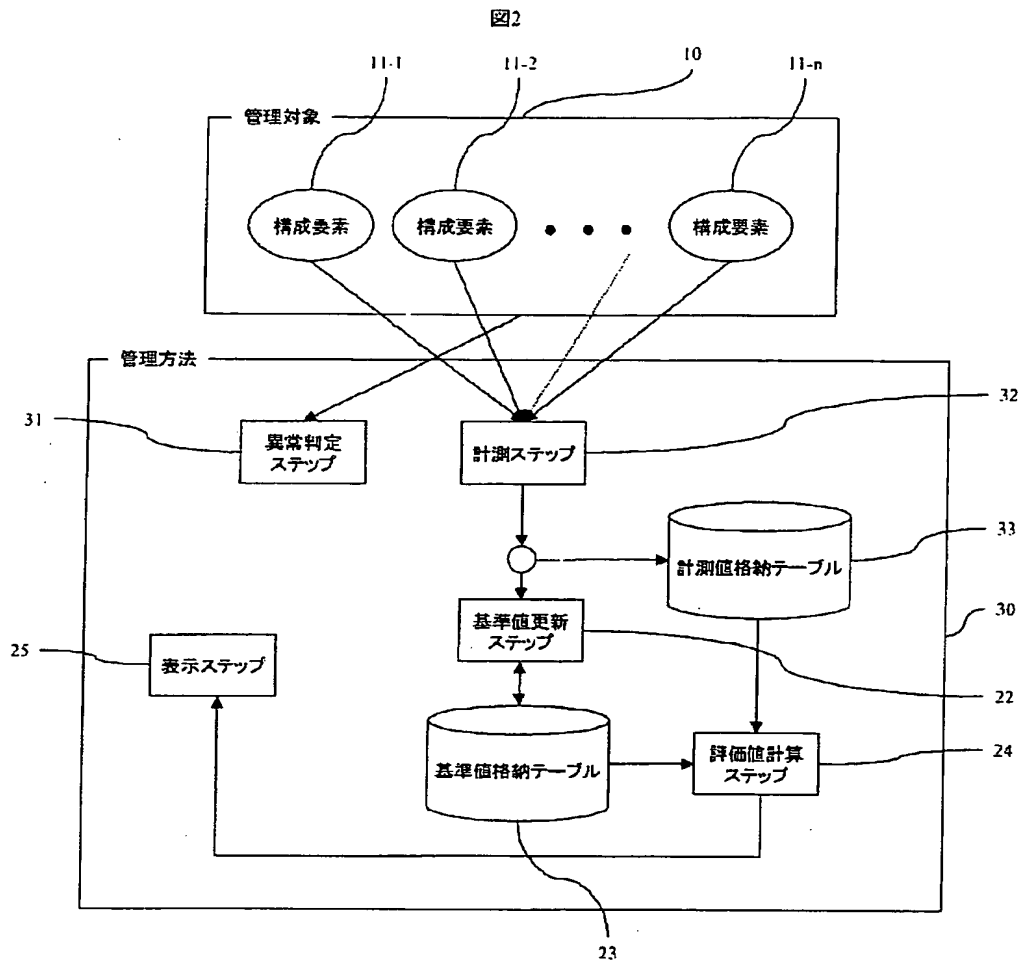
時刻	処理名	処理時間	基準処理時間	偏差 ▼
2000.5.29 10:16:50	プロセス C. 手続き γ1	10 秒	5 秒	5
2000.5.29 10:17:23	プロセス C. 手続き γ1	7 秒	5 秒	2
2000.5.29 10:17:15	プロセス R. 手続き β1	22 秒	20 秒	2
2000.5.29 10:16:11	プロセス B. 手続き β2	22 秒	22 秒	0
2000.5.29 10:16:38	プロセス A. 手続き α1	5 秒	6 秒	-1
2000.5.29 10:15:30	プロセス H. 手続き β1	18 秒	20 秒	-2

【図8】

図8

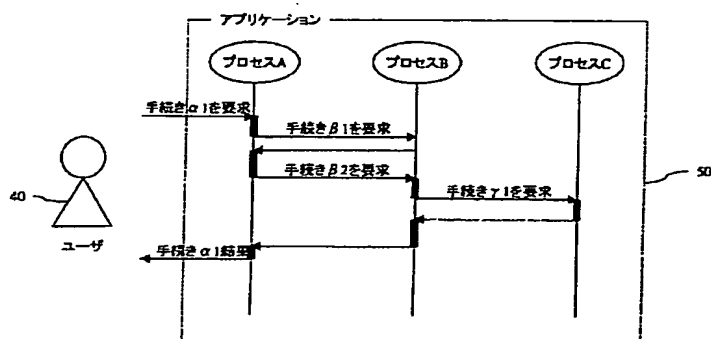
時刻	処理名	処理時間	基準処理時間	比率 ▼
2000.5.29 10:16:50	プロセス C. 手続き γ1	7 秒	5 秒	1.4
2000.5.29 10:16:11	プロセス H. 手続き β2	22 秒	22 秒	1
2000.5.29 10:16:30	プロセス B. 手続き β1	18 秒	20 秒	0.9
2000.5.29 10:16:38	プロセス A. 手続き α1	5 秒	6 秒	0.83

【図2】

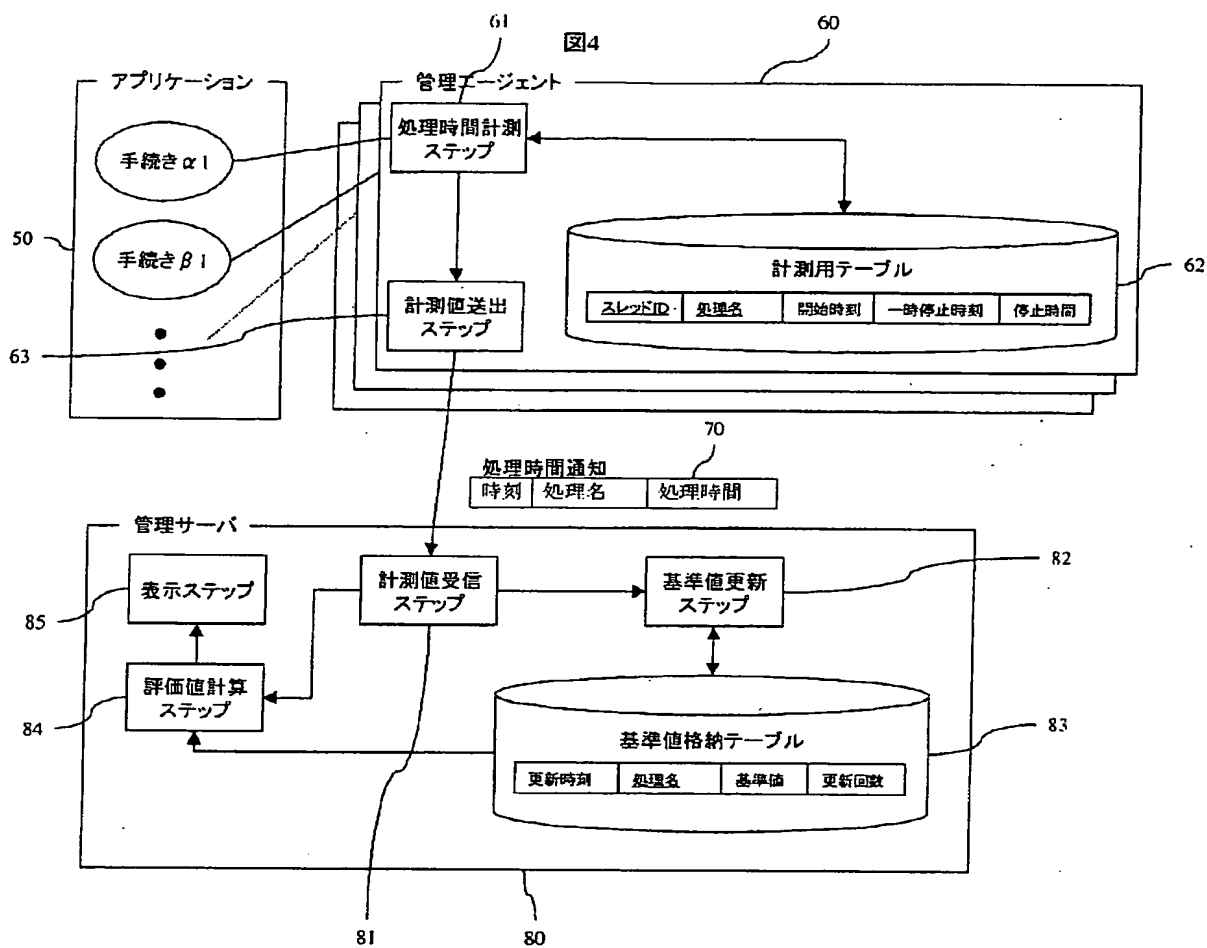


【図3】

図3

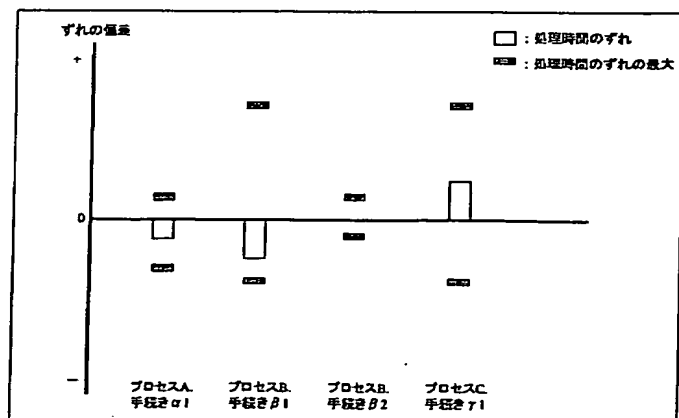


【図4】

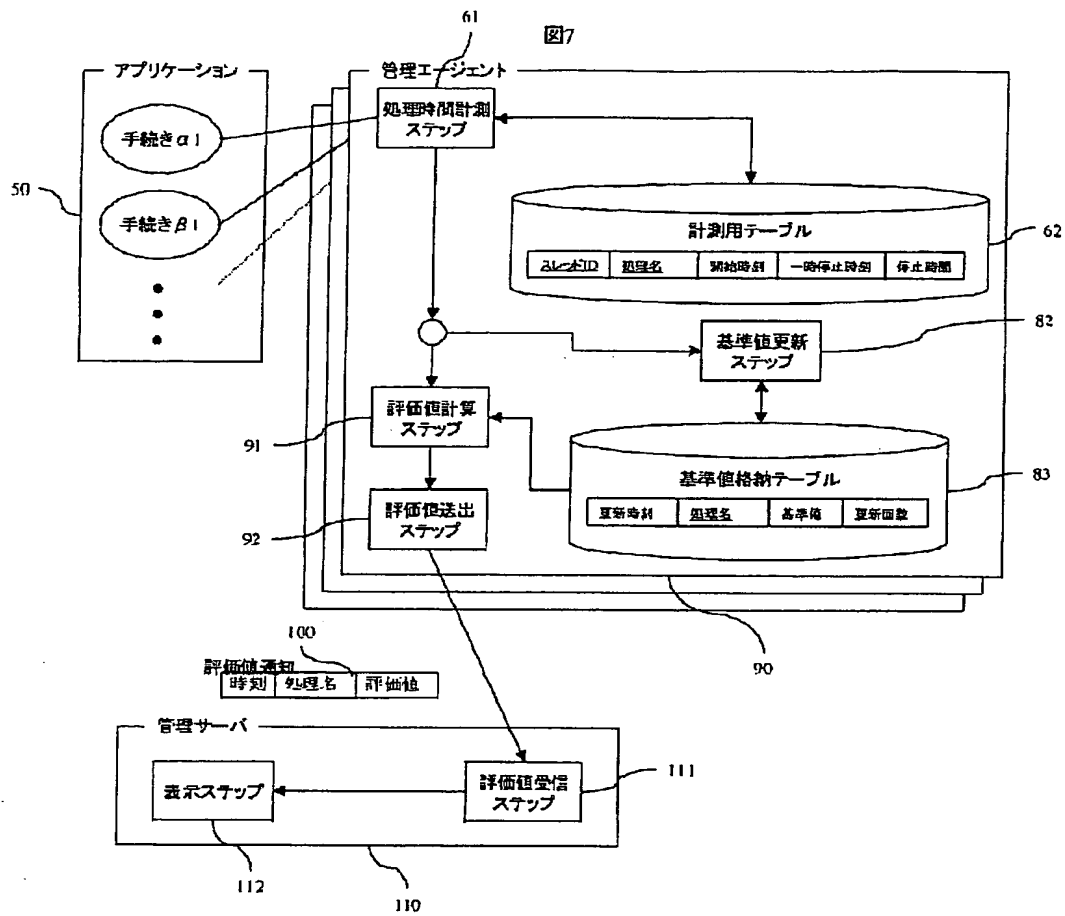


【図6】

図6

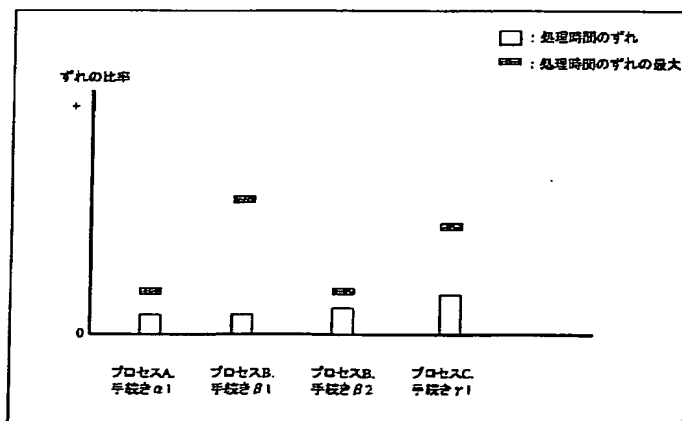


【図7】

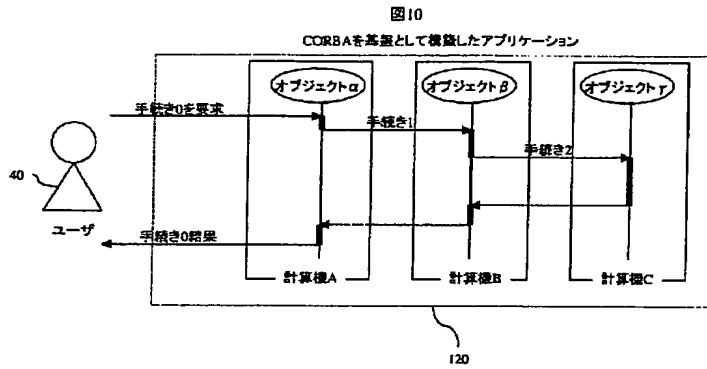


【図9】

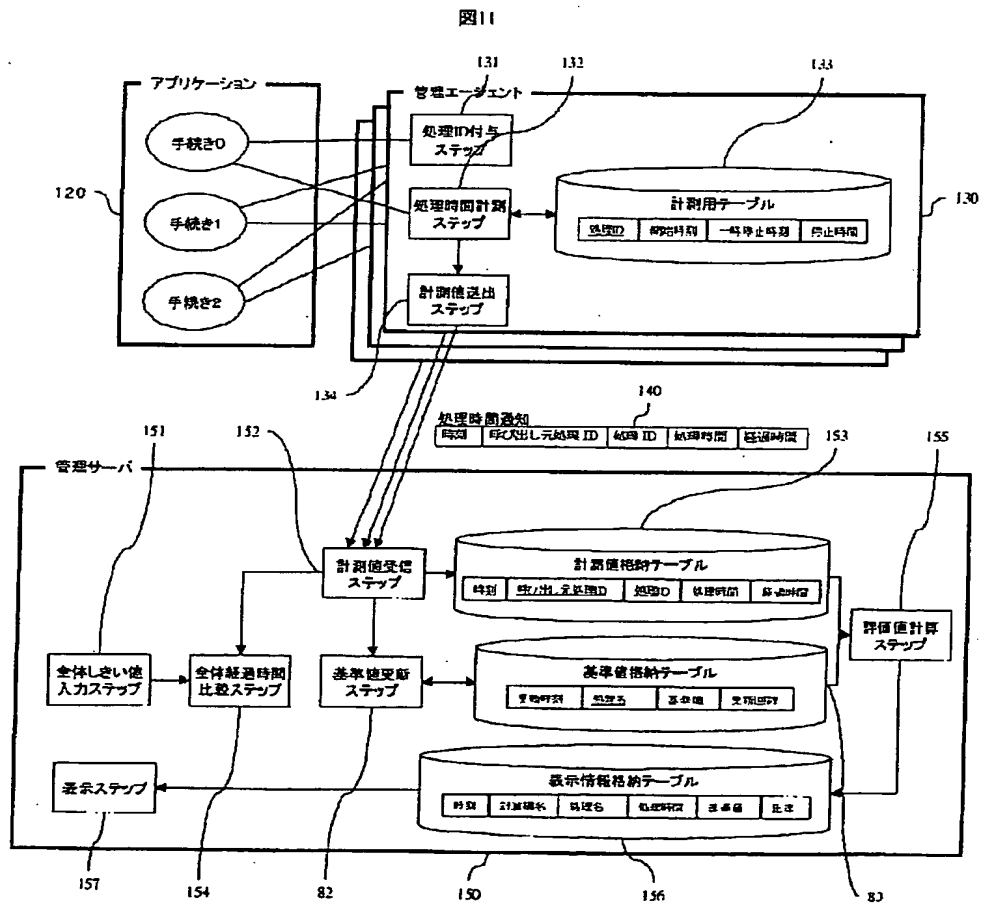
図9



【図10】

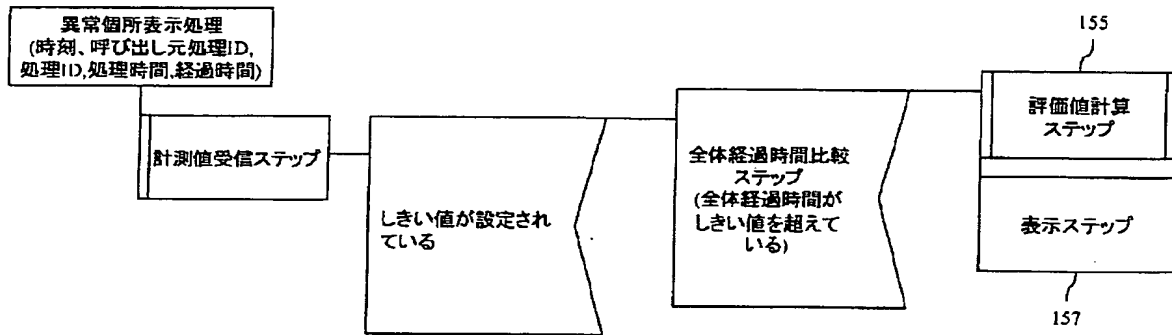


【図11】



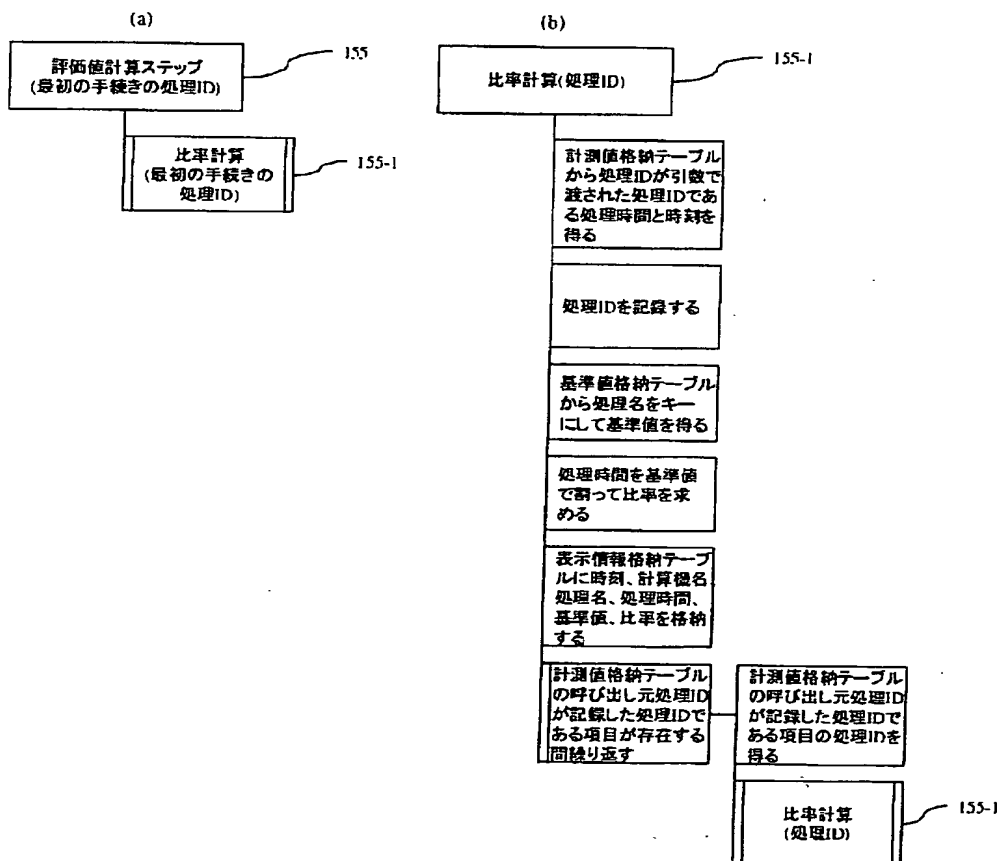
【図12】

図12

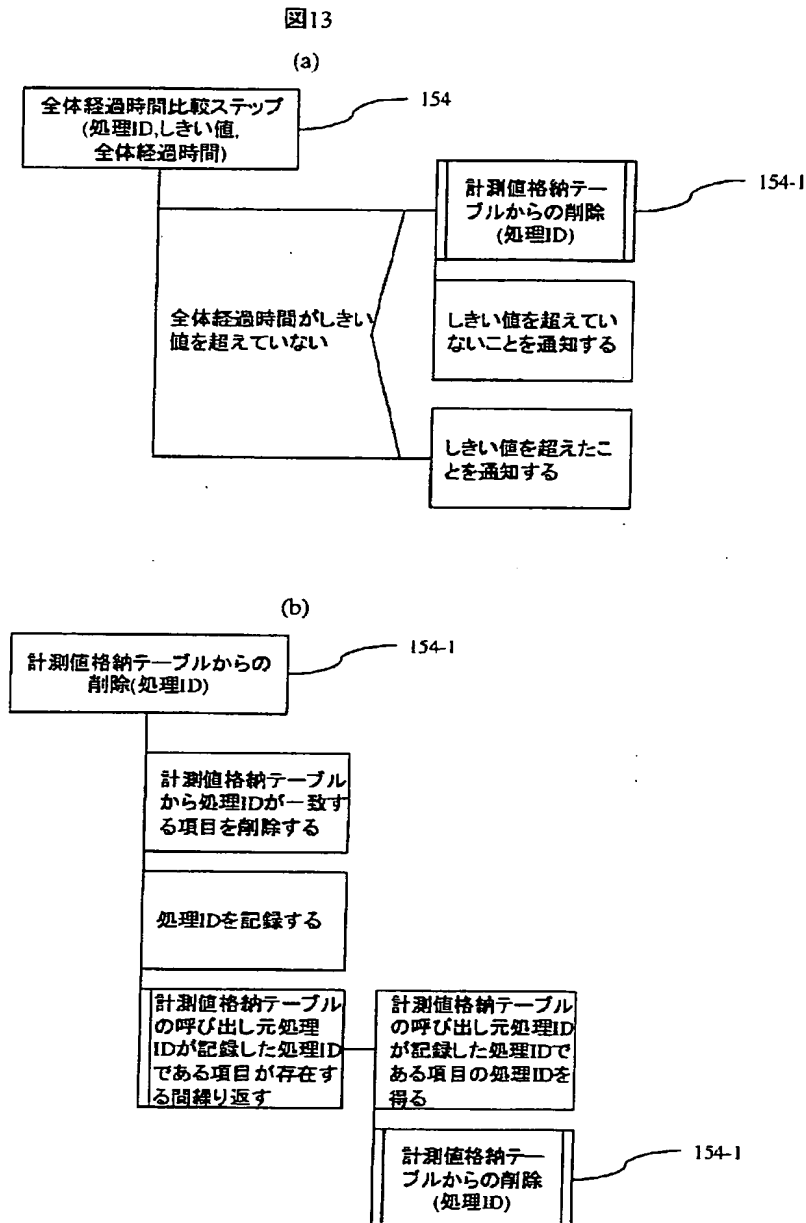


【図14】

図14



【図13】



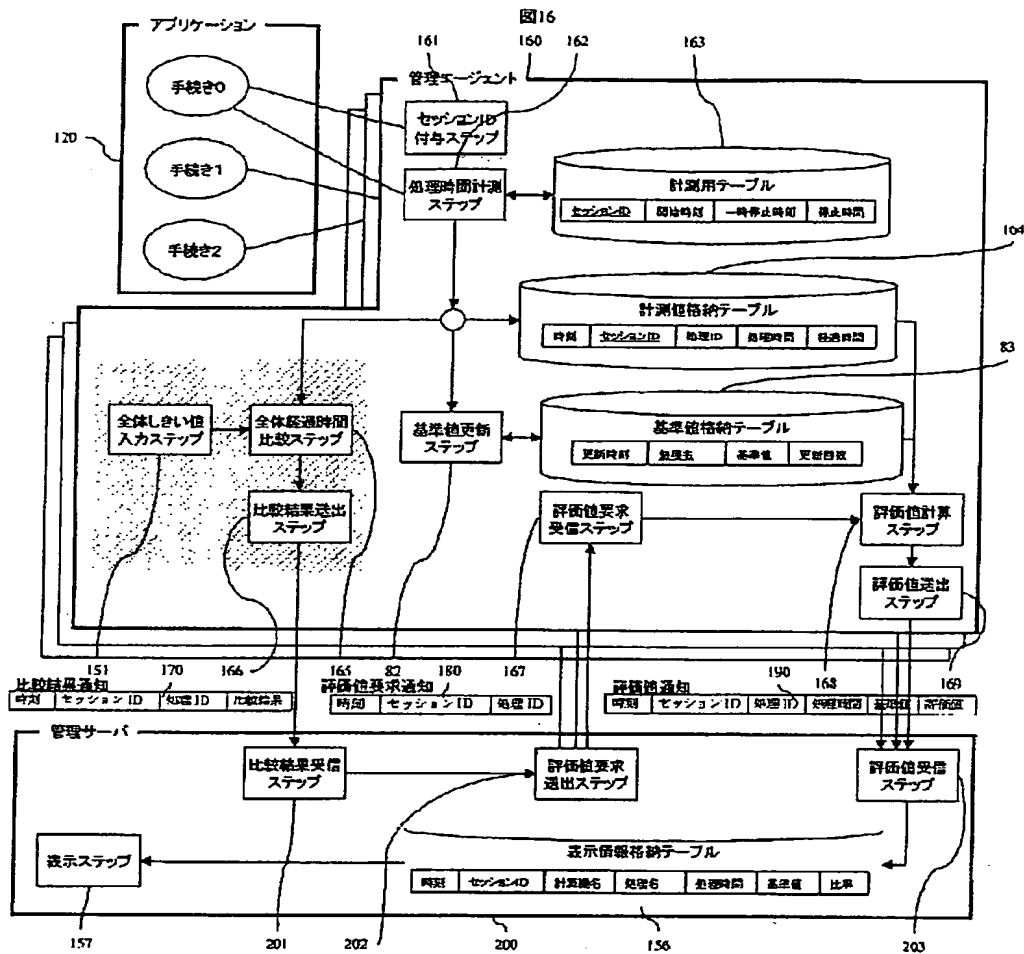
【図15】

図15

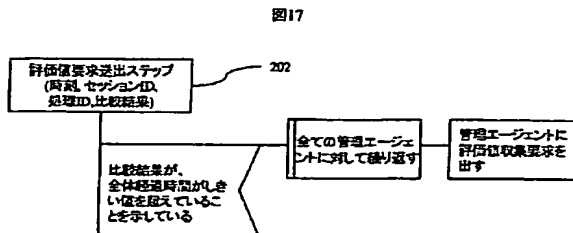
時刻	計算機名	処理名	処理時間	基準処理時間	比率▼
2000.5.29 10:15:32	計算機 C	オブジェクト γ. 手続き 2	19 秒	10 秒	1.9
2000.5.29 10:15:52	計算機 B	オブジェクト β. 手続き 1	16 秒	15 秒	1.07
2000.5.29 10:15:50	計算機 A	オブジェクト α. 手続き 0	27 秒	30 秒	0.9



【図16】

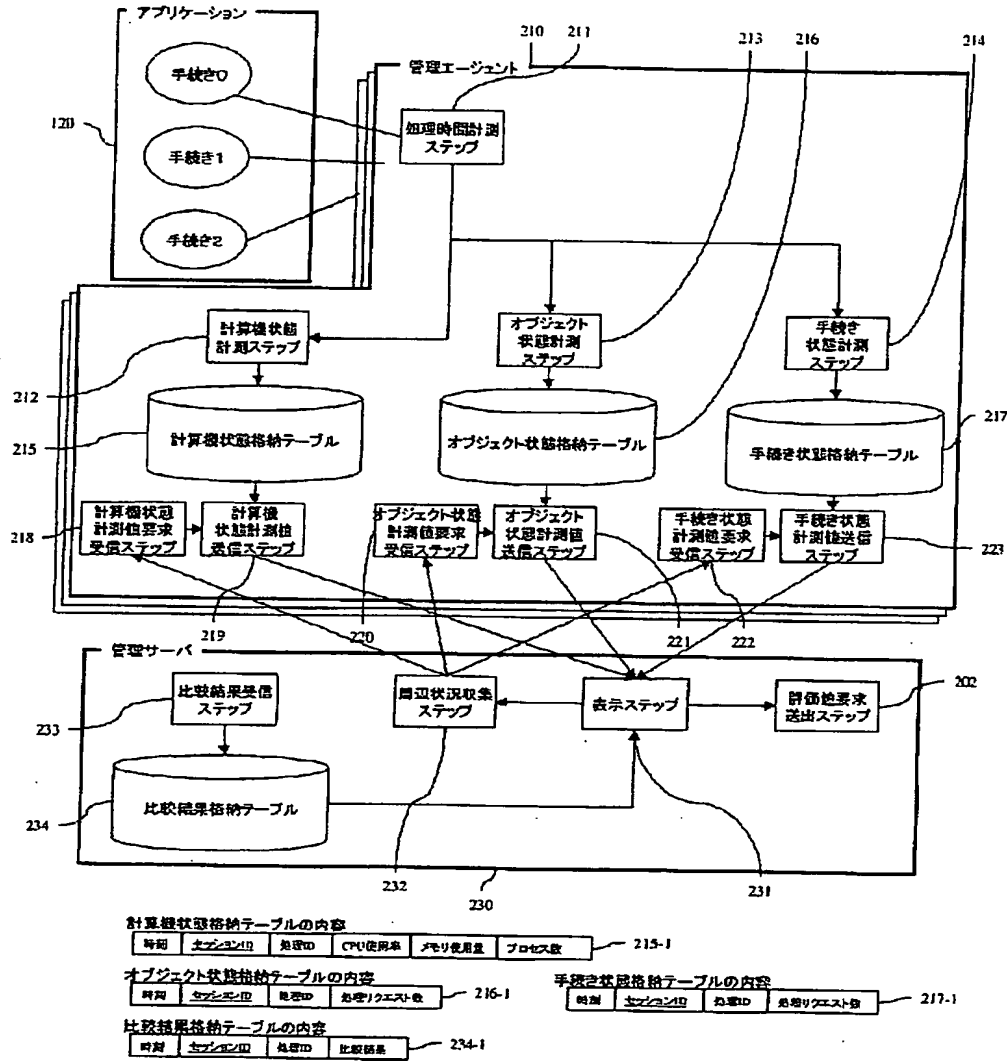


【図17】



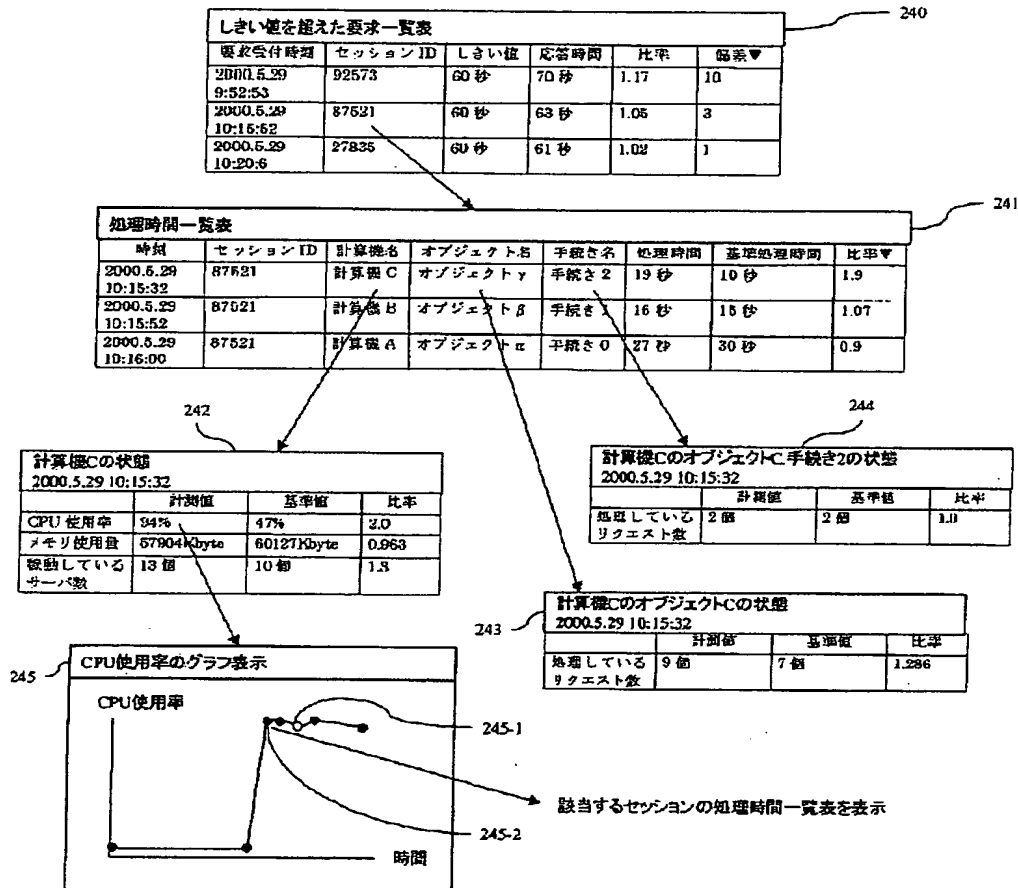
【図18】

図18



【図19】

図19



フロントページの続き

(72)発明者 黒田 沢希  
神奈川県川崎市麻生区王禅寺1099番地 株  
式会社日立製作所システム開発研究所内

(72)発明者 園村 智弘  
神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内

(72)発明者 藤井 直大  
神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内

Fターム(参考) 5B042 GA12 JJ23 JJ30 KK13 LA08  
MC28 MC33  
5B098 GA04 GC10 JJ08